

Generative Adversarial Networks(GANs)

Mansour DIA

20 août 2021

Goal

Discriminative vs Generative models

Generative Models vs. Discriminative Models

Discriminative models



Features Class
 $X \rightarrow Y$
 $P(Y|X)$

Generative models



Noise Class Features
 $\xi, Y \rightarrow X$
 $P(X|Y)$

FIGURE – GANs network structure.

GANs network structure

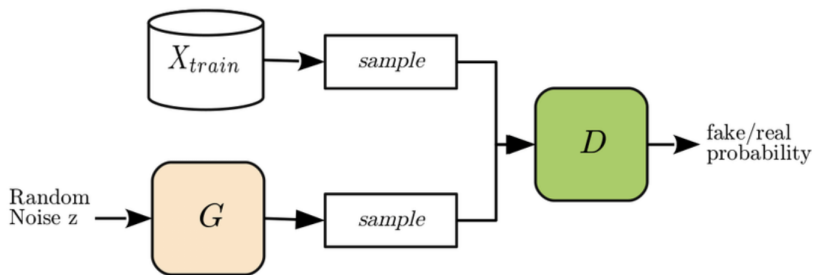
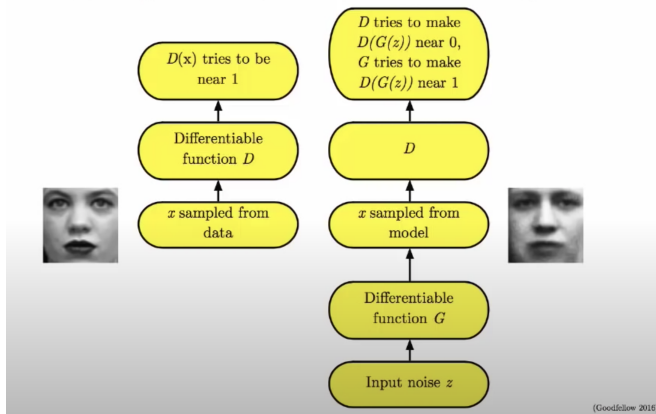


FIGURE – GANs network structure.

The Loss function

$$\min_G \max_D V(D, G) = \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} [\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})} [\log(1 - D(G(\mathbf{z})))]$$

Adversarial Nets Framework



Training loop

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k , is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**

for k steps **do**

- Sample minibatch of m noise samples $\{\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(m)}\}$ from noise prior $p_g(\mathbf{z})$.
- Sample minibatch of m examples $\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\mathbf{x})$.
- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^m \left[\log D(\mathbf{x}^{(i)}) + \log (1 - D(G(\mathbf{z}^{(i)}))) \right].$$

end for

- Sample minibatch of m noise samples $\{\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(m)}\}$ from noise prior $p_g(\mathbf{z})$.
- Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^m \log (1 - D(G(\mathbf{z}^{(i)}))).$$

end for

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

FIGURE – Training loop.

Random noise

- sampled from normal distribution,
- mean 0 and std 1.

The Generator

- 4 non linear block sequences ,

The Generator

- 4 non linear block sequences ,
- each block : linear, Batchnorm, ReLu ,

The Generator

- 4 non linear block sequences ,
- each block : linear, Batchnorm, ReLu ,
- Linear + Sigmoid.

The Discriminator

- 3 non linear block sequences ,

The Discriminator

- 3 non linear block sequences ,
- each block : linear, LeakyReLu ,

The Discriminator

- 3 non linear block sequences ,
- each block : linear, LeakyReLu ,
- Linear.

Final hyperparameters setting

- $n_epochs = 250$,
- $noise_vect_dim = 64$,
- $batch_size = 128$
- $lr = 0.00001$

Results

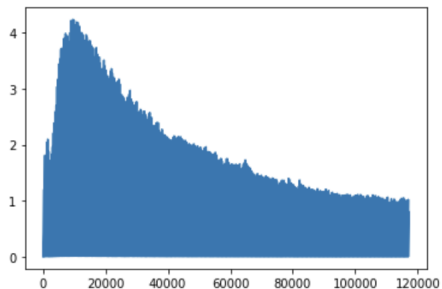


FIGURE – loss of the generator.

Results

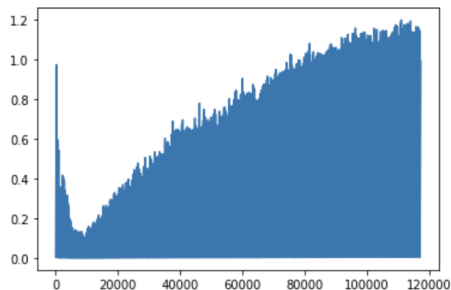


FIGURE – loss of the discriminator.

Results

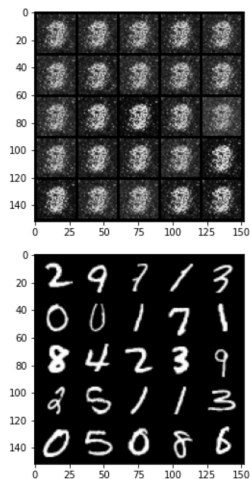


FIGURE – epoch 1.

Results

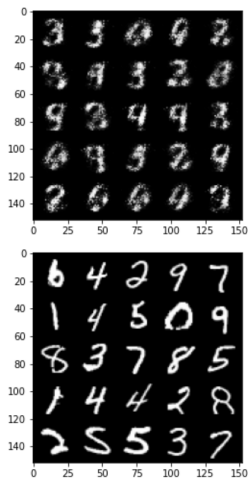


FIGURE – epoch 20.

Results

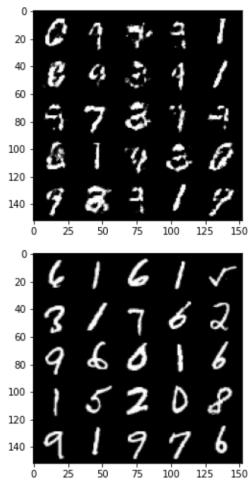


FIGURE – epoch 250.

Future work

- Convolutional layers insted of MLP ,

End

Thank you for your attention !